**Day 1: Introduction to DevOps**

Definition and history of DevOps

Why DevOps is important

DevOps principles and goals

DevOps culture and mindset

**Day 2: Version Control and Collaboration**

Introduction to Git and GitHub/GitLab/Bitbucket

Branching and merging strategies

Collaborative coding with version control

**Day 3: Continuous Integration (CI)**

CI concepts and benefits

Setting up a CI/CD pipeline

CI tools like Jenkins, Travis CI, or CircleCI

Writing and running automated tests

**Day 4: Continuous Deployment/Delivery (CD)**

Introduction to CD

Deployment strategies

Tools for CD like Docker, Kubernetes, and Helm

Implementing CD pipelines

**Day 5: Configuration Management**

Configuration management concepts

Tools like Ansible, Puppet, and Chef

Managing infrastructure as code

**Day 6: Monitoring and Logging**

Importance of monitoring in DevOps

Tools for monitoring (Prometheus, Grafana, Nagios)

Logging and log analysis (ELK stack, Splunk)

**Day 7: DevOps Best Practices and Culture**

DevOps best practices and patterns

Security in DevOps

DevOps culture and organizational alignment

Case studies and success stories

Throughout the training, you can include hands-on labs, exercises, and real-world scenarios to reinforce the concepts. Additionally, encourage participants to work on a project or case study during the training to apply what they've learned.

Remember that DevOps is a broad field, and a 7-day module can only cover the basics. Depending on the audience's prior knowledge and the specific goals of the training, you can adjust the content and depth accordingly. After the training, participants may want to explore advanced topics or pursue certifications in DevOps if they find it to be a valuable career path.

Reference link :

1. <https://www.youtube.com/watch?v=eX3ZimWWHh4&list=PLBGx66SQNZ8aPsFDwb79JrS2KQBTIZo10>
2. <https://www.youtube.com/watch?v=rKNSc8RrwxA&list=PL6XT0grm_TfgtwtwUit305qS-HhDvb4du>